



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:	Wu <i>et al.</i>)	
)	
Serial No.:	10/017,793)	Group Art Unit: 2183
)	
Filed:	December 12, 2001)	Examiner: Gerstl, Shane F.
)	
Title:	Run-Ahead Program)	
	Execution With Value)	
	Prediction)	

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Pursuant to the Notice of Appeal filed on April 13, 2005, Applicants (hereafter "Appellants") hereby submit this Appeal Brief in support of an Appeal from the Final Decision by the Examiner in the above-captioned patent application. Appellant respectfully requests consideration of this Appeal by the Board of Patent Appeals and Interferences for allowance of the claims in the above-captioned patent application.

It is not believed that extensions of time are required beyond those that may otherwise be provided for in documents accompanying this Appeal. However, if additional extensions of time are necessary to prevent abandonment of this application, then such extensions of time are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required therefore are hereby authorized to be charged to Deposit Account No. 02-2666.

10/17/2005 HDESTA1 00000119 10017793

01 FC:1402
02 FC:1254

500.00 DP
~~1500.00 DP~~

I. Real Party in Interest

The real party in interest is the assignee of the full interest in the invention, Intel Corporation of 2200 Mission College Boulevard, Santa Clara, California 95054-1549.

II. Related Appeals and Interferences

To the best of Appellants' knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision in the instant appeal.

III. Status of Claims

Claims 1-27 are pending in the application and were finally rejected in an Office Action mailed January 13, 2005. Claims 1-27 are the subject of this appeal. A copy of Claims 1-27 as they stand on appeal are set forth in the Claims Appendix (Appendix A).

IV. Status of Amendments

In response to the first Office Action, dated July 28, 2004, an amendment was filed on October 28, 2004. No claim amendments were made in response to the Final Office Action, dated January 13, 2005. Thus, the attached Claims Appendix reflects the status of the claims listed in the amendment filed on October 28, 2004.

V. Summary of Claimed Subject Matter

The invention relates to an apparatus and methods for speeding up the processing of data. In particular, the invention relates to a run-ahead program execution that uses

value prediction to increase performance within a data processing environment. Independent claims 1 and 13 recite a data processing apparatus and a computer, respectively. As recited in independent claims 1 and 13, the data processing apparatus and the computer include a first pipeline (FIG. 1, item 105) having a data cache (FIG. 1, item 110) and an instruction cache (FIG. 1, item 115). (FIG. 1, items 105, 110, and 115; Specification, page 5, lines 7-10.) The apparatus and computer also include a second pipeline (FIG. 1, item 120) coupled to the data cache (FIG. 1, item 110) and the instruction cache (FIG. 1, item 115). (FIG. 1, items 120, 110, and 115; Specification, page 5, lines 10-13.) The apparatus and computer also include a data value prediction module (FIG. 1, item 150) coupled to the second pipeline (FIG. 1, item 120). (FIG. 1, items 150 and 120; Specification page 6, lines 3-4.)

Independent claims 20 and 24 are Beauregard and method claims directed to processing data. A plurality of instructions including a LOAD instruction are executed using a first pipeline sharing an instruction cache and a data cache with a second pipeline. (FIG. 2, method 280, block 282; Specification, page 9, lines 16-22.) A predicted load value for execution of the LOAD instruction is calculated if a cache miss in the data cache results when the second pipeline executes the LOAD instruction before the first pipeline. (FIG. 2, blocks 288 and 290; Specification, page 10, lines 4-6.) Execution of the plurality of instructions using the second pipeline are continued. (FIG. 2, block 287; Specification, page 10, line 6.)

With respect to dependent claim 4, the data processing apparatus also includes a first register file (FIG. 1, item 125) coupled to the first pipeline (FIG. 1, item 105) and a

second register file (FIG. 1, item 130) coupled to the second pipeline (FIG. 1, item 120). (FIG. 1, items 125, 105, 130, and 120; Specification, page 5, lines 24-25.)

With regards to dependent claim 7, the data cache (FIG. 1, item 110), the instruction cache (FIG. 1, item 115), and the data value prediction module (FIG. 1, item 150) are included in a single processor. (FIG. 1, items 110, 115, and 150; Specification page 8, lines 26-28; see also, Specification, page 8, lines 21-26.)

With regards to dependent claim 9, the data processing apparatus also includes a main memory (FIG. 1, item 122) coupled to the data cache (FIG. 1, item 110). (FIG. 1, items 122 and 110; Specification page 5, lines 18-19.) The first pipeline (FIG. 1, item 105) may operate to store a data value to the main memory (FIG. 1, item 122). (FIG. 1, items 105 and 122; Specification, page 5, lines 22-23.) The second pipeline (FIG. 1, item 120) may not operate to store the data value to the main memory (FIG. 1, item 122). (FIG. 1, items 120 and 122; Specification, page 5, lines 21-22.)

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1, 4, 7, 9, 13, 20, and 24 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,757,811 to Mukherjee.

VII. Argument

A. Claims 1, 4, 7, 9, 13, 20, and 24 are not anticipated by U.S. Patent No. 6,757,811 to Mukherjee

In the Final Office Action dated January 13, 2005, the Examiner stated, with regards to independent claims 1, 13, 20, and 24, and dependent claims 4, 7, and 9, that Mukherjee teaches every element listed in the claims. Final Office Action dated January 13, 2005, pp. 2-9. The Examiner maintained his rejection in the Advisory Action dated May 16, 2005.

To anticipate a claim of a pending application, a single reference must disclose each and every element of the claimed invention. *Hybritech Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 1397 (Fed. Cir. 1986). The exclusion of a claimed element from the single source is enough to negate anticipation by that reference. *Atlas Powder Co. v. E.I. du Pont de Nemours & Co.*, 750 F.2d 1569, 1574 (Fed. Cir. 1984).

Claims 1 and 13

With respect to independent claim 1, the Examiner states that Mukherjee teaches every element of Applicants' claimed invention. Applicants respectfully disagree.

Contrary to the present invention, Mukherjee does not teach or suggest every element of Applicants' invention. For example, referring to independent claim 1,

Mukherjee does not teach or suggest at least the following claimed element of “a data value prediction module coupled to the second pipeline.”

The present invention teaches a method of processing data using a main pipeline and a run-ahead pipeline. *Specification*, page 9, lines 16-24. Each pipeline begins execution of a program, the program containing a plurality of instructions. *Id.* The method proceeds with the run-ahead pipeline fetching an instruction. *Id.* at lines 25-26. If the fetched instruction is a LOAD instruction, and if it is determined that executing the LOAD instruction will produce a data cache miss, then a load value will be predicted using the data value prediction module. *Id.* at p. 10, lines 4-6. Thus, instead of waiting for the actual value to be retrieved from main memory when a data cache miss occurs, the run-ahead pipeline continues execution of the run-ahead thread using the predicted value. *Specification*, p. 4, lines 17-21. The predicted values are never stored to memory or the data cache. *Specification*, p. 4, line 31 – page 5, line 4; p. 6, lines 20-30.

Unlike the present invention, Mukherjee does not teach a value prediction module, and is silent on making a prediction value when a data cache miss occurs. *See also*, Mukherjee’s block diagram of the simultaneous and redundantly threaded processor in FIG. 2, which does not include a value prediction module. Instead, Mukherjee teaches that:

Cache misses occur when an instruction requests data from memory that is not also available in cache memory. The processor first checks whether the requested data already resides in the faster access cache memory, which generally is onboard the processor die. If the requested data is not present in cache (a condition referred to as a cache “miss”), then the processor is forced to retrieve the data from main system memory which takes more time, thereby causing latency, than if the data could have been retrieved from the faster onboard cache.

Mukherjee, col. 2, line 65 – col. 3, line 7.

Mukherjee solves the latency problem by using a simultaneous and redundantly threaded processor that simultaneously executes the same program in two separate threads. *Id.* at col. 3, line 65 – col. 4, line 2. This is accomplished by processing one thread ahead of the other, thus creating a “slack” of instructions between the two threads so that instructions in one thread are processed ahead of the other thread. *Id.* at col. 4, lines 11-16. The slack allows all or at least some of the cache misses or branch misspeculations encountered by the leading thread to be resolved before the corresponding instructions from the trailing thread are fetched. *Id.* at col. 4, lines 19-24.

Thus, unlike the present invention, which uses a data value prediction module to predict a value when a data cache miss occurs, Mukherjee teaches that the correct value from main memory is loaded into the data cache when a data cache miss occurs. *See Mukherjee*, col. 7, lines 53-55, stating that “[s]ome or all cache misses in the leading thread will result in the requested data being written to the processor’s data cache.” Thus, with Mukherjee, a cache miss processed in the leading thread eliminates a corresponding cache miss in the trailing thread by retrieving the requested data from the main memory during processing of the leading thread. *Id.* In fact, Mukherjee teaches away from Applicants’ element of “a data value prediction module coupled to the second pipeline” because Mukherjee allows the leading thread to resolve the cache miss by enabling the requested data to be written to the processor’s data cache so that the corresponding instructions in the trailing thread will not experience the cache miss. Thus, with Mukherjee there is no need to have a data value prediction module because the missing cache value is retrieved from main memory.

For at least these reasons, Applicants respectfully submit that Mukherjee does not include each and every element of Applicants' claimed invention as recited in independent claim 1. Independent claim 13 recites similar elements to claim 1. Therefore, independent claims 1 and 13, and the claims that depend therefrom (claims 2-12 and 14-19, respectively), are not anticipated by Mukherjee.

Claims 20 and 24

With regards to claim 20, Mukherjee does not teach or suggest at least the following element of "calculating a predicted load value for execution of the LOAD instruction if a cache miss in the data cache results when the second pipeline executes the LOAD instruction before the first pipeline." Unlike the present invention, which supplies a predicted value for the load value instead of an actual value retrieved from memory, Mukherjee does not need to predict a load value because Mukherjee actually resolves the cache miss with the leading thread by retrieving the actual value from memory and storing it in the cache so that the corresponding instructions in the trailing thread will not experience the cache miss.

Thus, for at least these reasons, Applicants respectfully submit that Mukherjee does not include each and every element of Applicants' claimed invention as recited in independent claim 20. Independent claim 24 includes similar elements to claim 20. Therefore, independent claims 20 and 24, and the claims that depend therefrom (claims 21-23 and 25-27, respectively), are not anticipated by Mukherjee.

Claim 4

With respect to dependent claim 4, Mukherjee does not teach or suggest a first register file coupled to the first pipeline; and a second register file coupled to the second pipeline. Instead, Mukherjee teaches a floating point register and an integer register that is used by both threads depending on whether the instruction is a floating point instruction or an integer instruction. *Mukherjee*, col. 6, lines 25-30. Thus, unlike the present invention, which uses a first register file coupled to the first pipeline and a second register file coupled to the second pipeline, Mukherjee uses a floating point register and an integer register that is designated for use by both threads. Thus, for at least this reason, claim 4 is not anticipated by Mukherjee.

Claim 7

With respect to dependent claim 7, Mukherjee does not teach that the data cache, the instruction cache, and the data value prediction module are included in the single processor. As indicated above, Mukherjee does not teach or suggest a data value prediction module, and therefore, cannot teach that the data cache, the instruction cache, and the data value prediction module are included in the single processor. *See* Mukherjee, FIG. 2, which does not show a data value prediction module included in the processor. Thus, for at least this reason, claim 7 is not anticipated by Mukherjee.

Claim 9

With respect to dependent claim 9, Mukherjee does not teach that the leading thread (referred to as the second pipeline in the invention) may not operate to store the

data value to the main memory. In fact, Mukherjee is silent on whether the leading thread may not operate to store the data value to the main memory. Thus, for at least this reason, claim 9 is not anticipated by Mukherjee.

Atty. Docket No.: 42P12589

Wu *et al.*
Appl. No. 10/017,793

Conclusion

In view of the foregoing, favorable reconsideration and reversal of the rejections is respectfully requested. Early notification of the same is earnestly solicited. If there are any questions regarding the present application, the Examiner and/or the Board is invited to contact the undersigned attorney at the telephone number listed below.

Respectfully submitted,

Intel Corporation

Dated: October 13, 2005

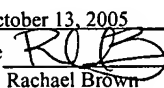
/Crystal D. Sayles, Reg. No. 44,318/
Crystal D. Sayles
Senior Attorney
Intel Americas, Inc.
(202) 986-3179

c/o Blakely, Sokoloff, Taylor & Zafman, LLP
12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025-1026

I hereby certify that this correspondence is being deposited with the United States Postal service as first class mail with sufficient postage in an envelope addressed to:
Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450

On: October 13, 2005

Signature


Rachael Brown

Date

10/13/05

Appendix A: Claims Appendix

1. (original) A data processing apparatus, comprising:
 - a first pipeline having a data cache and an instruction cache;
 - a second pipeline coupled to the data cache and the instruction cache; and
 - a data value prediction module coupled to the second pipeline.
2. (original) The data processing apparatus of claim 1, further comprising:
 - a first instruction fetch module coupled to the first pipeline; and
 - a second instruction fetch module coupled to the second pipeline.
3. (original) The data processing apparatus of claim 2, further comprising:
 - a branch predictor coupled to the first and second instruction fetch modules.
4. (original) The data processing apparatus of claim 1, further comprising:
 - a first register file coupled to the first pipeline; and
 - a second register file coupled to the second pipeline.
5. (original) The data processing apparatus of claim 1, wherein the first pipeline is included in a first processor, and wherein the second pipeline is included in a second processor.
6. (original) The data processing apparatus of claim 1, wherein the first and second pipelines are included in a single processor.
7. (original) The data processing apparatus of claim 6, wherein the data cache, the instruction cache, and the data value prediction module are included in the single processor.
8. (original) The data processing apparatus of claim 1, further comprising:
 - a value prediction table coupled to the value prediction module.
9. (original) The data processing apparatus of claim 1, further comprising:
 - a main memory coupled to the data cache, wherein the first pipeline may operate to store a data value to the main memory, and wherein the second pipeline may not operate to store the data value to the main memory.
10. (original) The data processing apparatus of claim 1, further comprising:
 - a storage buffer coupled to the second pipeline.
11. (original) The data processing apparatus of claim 1, further comprising:
 - a synchronization mechanism coupled to the second pipeline.

12. (original) The data processing apparatus of claim 11, wherein the synchronization mechanism includes a misprediction counter.
13. (original) A computer, comprising:
 - a first processor including a first pipeline having a data cache coupled to a memory, and an instruction cache;
 - a second pipeline coupled to the data cache and the instruction cache; and
 - a data value prediction module coupled to the second pipeline.
14. (original) The computer of claim 13, further comprising:
 - a second processor including the second pipeline.
15. (original) The computer of claim 13, further comprising:
 - a bus coupled to the data cache and the memory, wherein the first processor included the second pipeline.
16. (original) The computer of claim 13, further comprising:
 - a value prediction table coupled to the value prediction module.
17. (original) The computer of claim 13, further comprising:
 - a synchronization mechanism coupled to the second pipeline.
18. (original) The computer of claim 17, wherein the synchronization mechanism includes a run-ahead counter.
19. (previously presented) The computer of claim 13, further comprising:
 - a storage buffer coupled to the second pipeline.
20. (previously presented) An article comprising a computer-readable medium having associated data, wherein the medium causes a computer to perform the following:
 - executing a plurality of instructions including a LOAD instruction using a first pipeline sharing an instruction cache and a data cache with a second pipeline;
 - calculating a predicted load value for execution of the LOAD instruction if a cache miss in the data cache results when the second pipeline executes the LOAD instruction before the first pipeline; and
 - continuing execution of the plurality of instructions using the second pipeline.
21. (previously presented) The article of claim 20, wherein the computer-readable medium further causes the computer to perform the following:
 - counting a number of mispredictions occurring when the predicted load value is incorrect; and

restarting execution of the plurality of instructions by the second pipeline at a program counter value maintained by the first pipeline if the number of mispredictions is greater than or equal to a preselected threshold value.

22. (previously presented) The article of claim 20, wherein the computer-readable medium further causes the computer to perform the following:

counting a number of instructions included in the plurality of instructions which the second pipeline has executed ahead of the first pipeline; and

restarting execution of the plurality of instructions by the second pipeline at a program counter value maintained by the first pipeline if the number of instructions is greater than or equal to a preselected threshold value.

23. (previously presented) The article of claim 20, wherein the computer-readable medium further causes the computer to perform the following:

beginning execution of the plurality of instructions by the first and second pipelines at a same program counter value.

24. (original) A method of processing data, comprising:

executing a plurality of instructions including a LOAD instruction using a first pipeline sharing an instruction cache and a data cache with a second pipeline;

calculating a predicted load value for execution of the LOAD instructions if a cache miss in the data cache results when the second pipeline executes the LOAD instruction before the first pipeline; and

continuing execution of the plurality of instructions using the second pipeline.

25. (original) The method of claim 24, further comprising:

counting a number of mispredictions occurring when the predicted load value is incorrect; and

restarting execution of the plurality of instructions by the second pipeline at a program counter value maintained by the first pipeline if the number of mispredictions is greater than or equal to a preselected threshold value.

26. (original) The method of claim 24, further comprising:

counting a number of instructions included in the plurality of instructions which the second pipeline has executed ahead of the first pipeline; and

restarting execution of the plurality of instructions by the second pipeline at a program counter value maintained by the first pipeline if the number of instructions is greater than or equal to a preselected threshold value.

27. (original) The method of claim 24, further comprising:

beginning execution of the plurality of instructions by the first and second pipelines at a same program counter value.

Appendix B: Evidence Appendix

No evidence has been submitted in the present appeal.

Appendix C: Related Proceedings Appendix

There are no related proceedings.